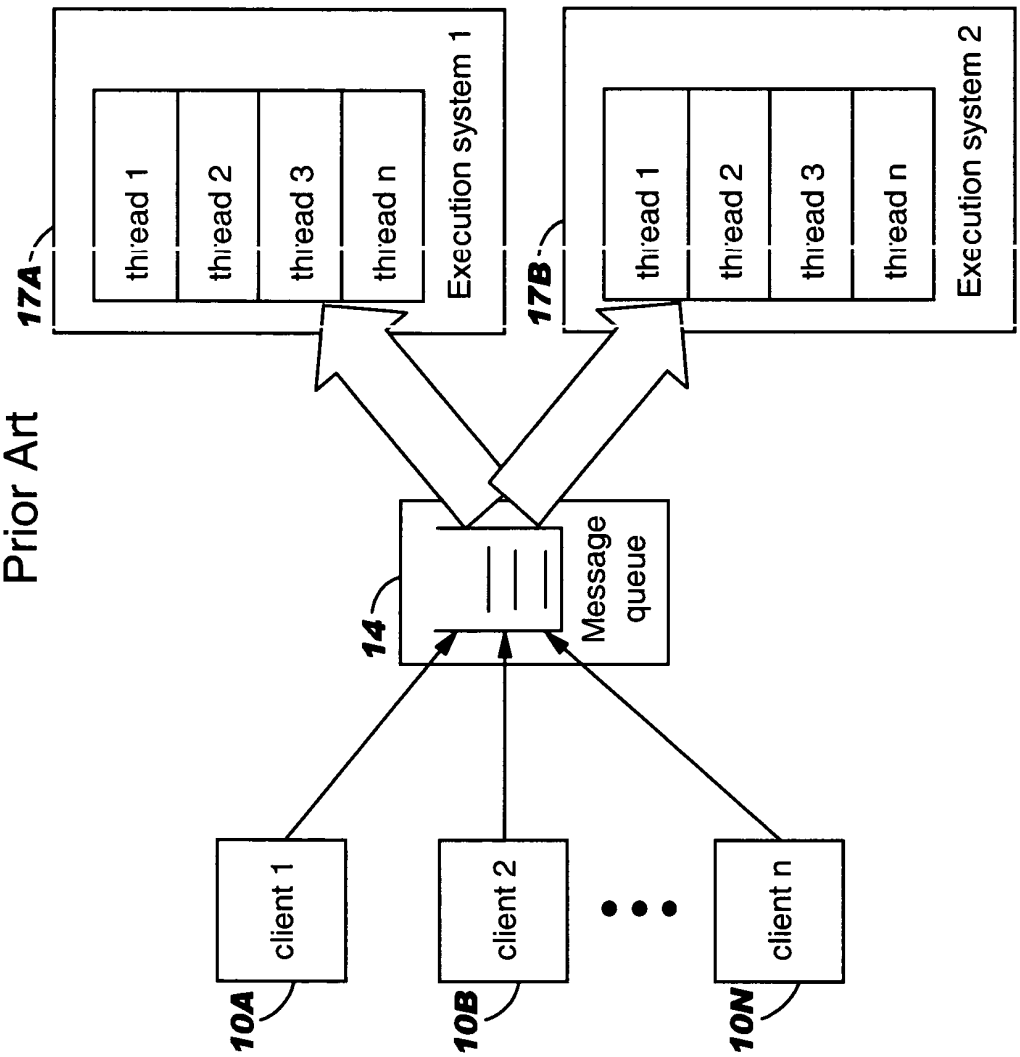


FIG. 1
Prior Art

FIG. 2

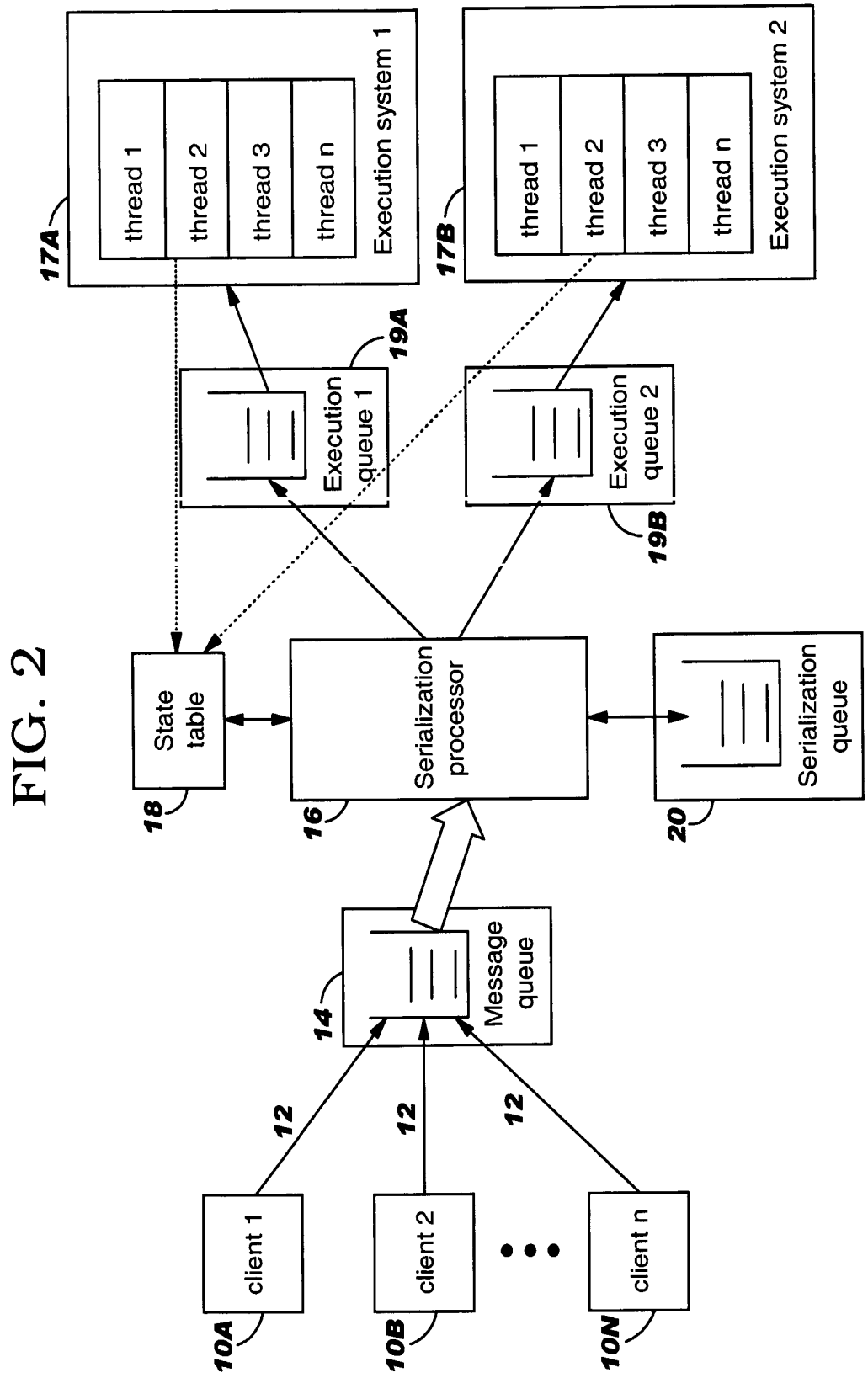

Execution system 1
thread 1
thread 2
thread 3
thread n
17A
Execution system 2
thread 1
thread 2
thread 3
thread n
17B
Execution queue 1
19A
Execution queue 2
19B
State table
18
Serialization processor
16
Serialization queue
20
Message queue
14
12
12
12
client 1
client 2
client n
10A
10B
10N

FIG. 3

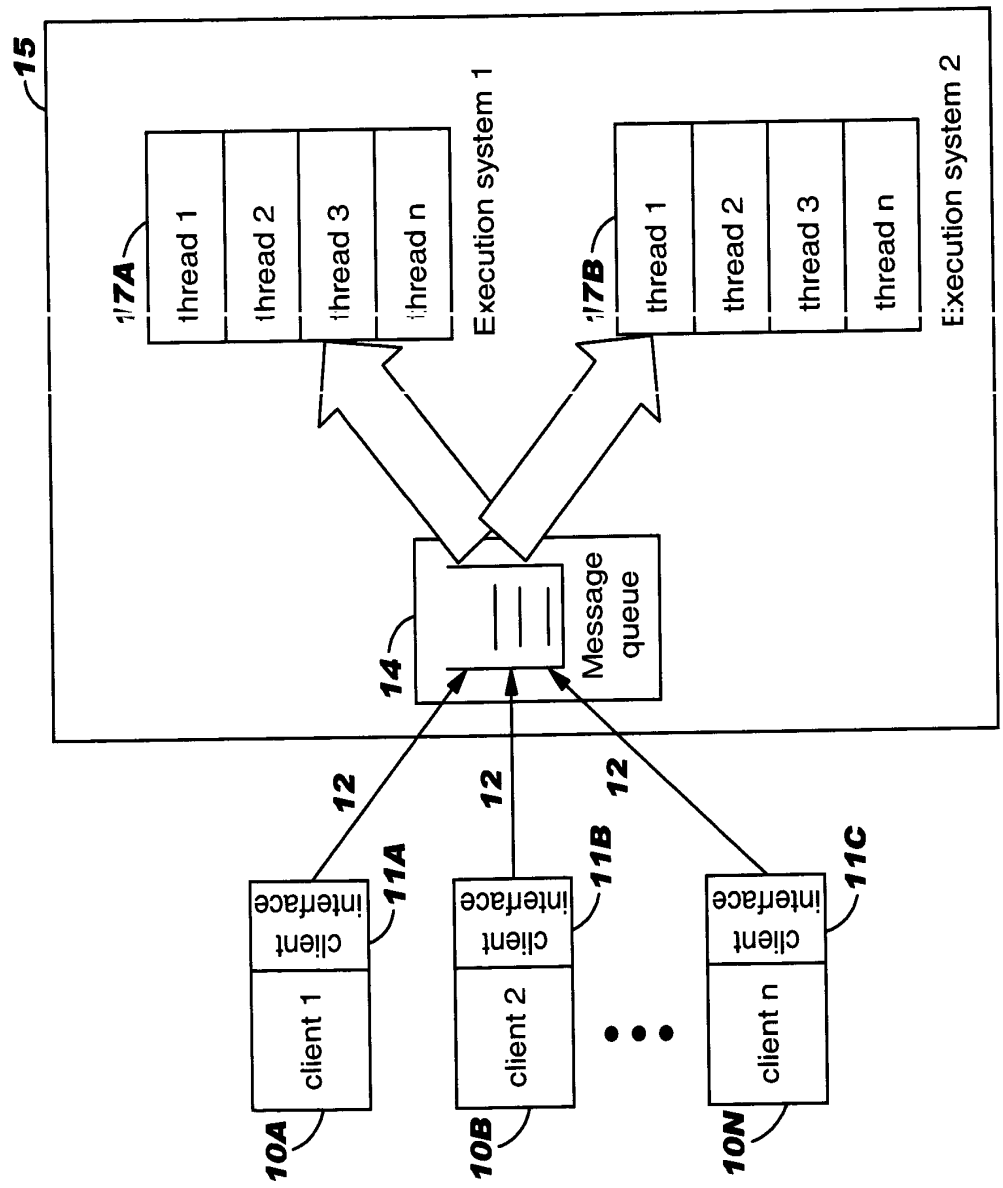

15
17A
thread 1
thread 2
thread 3
thread n
Execution system 1
17B
thread 1
thread 2
thread 3
thread n
Execution system 2
14
Message queue
12
12
12
client interface
11A
11B
11C
client 1
client 2
client n
10A
10B
10N

FIG. 4

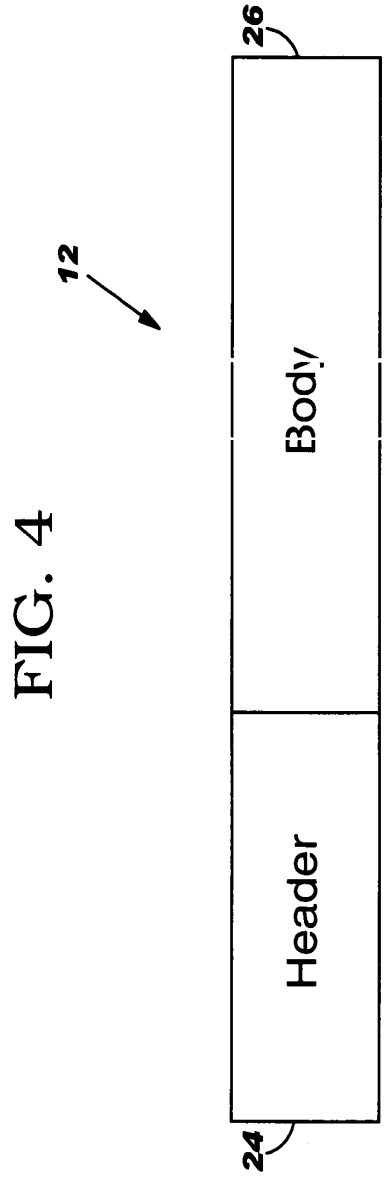


FIG. 5
Prior Art

24

MQMD name	type	value
StrucId	MQCHAR4	MQMD_STRUC_ID(Default)
Version	MQLONG	MQMD_VERSION_1(Default)
Report	MQLONG	tbd
MsgType	MQLONG	
Expiry	MQLONG	Default = UNLIMITED
Feedback	MQLONG	tbd
Encoding	MQLONG	native
CodedCharSetId	MQLONG	Same as queue manager (Default
Format	MQCHAR8	STRING
Priority	MQLONG	Specify a value between zero and 16.
Persistence	MQLONG	NOT_PERSISTENT
MsgId	MQBYTE24	Generated by queue manager (default)
CorrelId	MQBYTE24	NONE (default)
BackoutCount	MQLONG	Initial value = 0
ReplyToQ	MQCHAR48	Queue name for a reqply (none is default)
ReplyToQMgr	MQCHAR48	Queue manager name for a reqply (none is default)
UserIdentifier	MQCHAR12	userid of user which has logged on to the application (optional)
AccountingToken	MQBYTE32	Set to hex 0.
ApplIdentityData	MQCHAR32	Used for development purposes.
PutApplType	MQLONG	Set by Queue Manager (optional)
PutApplName	MQCHAR24	Set by Queue Manager (optional)
PutDate	MQCHAR8	Set by Queue Manager (optional)
PutTime	MQCHAR8	Set by Queue Manager (optional)
ApplOriginData	MQCHAR4	Set to blank by Queue Manager

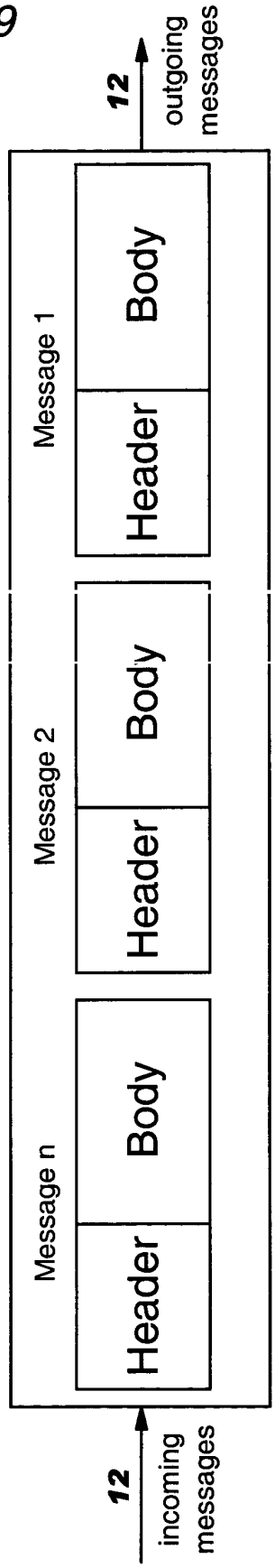

12
outgoing
messages
Message 1
Header
Body
Message 2
Header
Body
Message n
Header
Body
12
incoming
messages
14

FIG. 6

FIG. 7

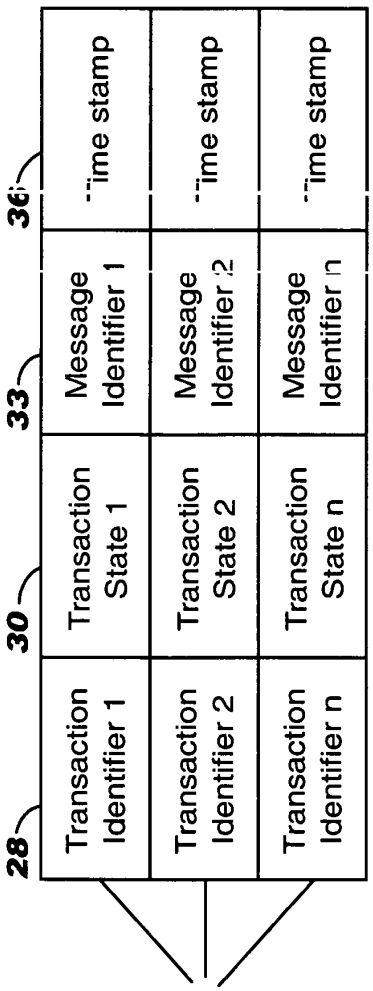

18
28
30
33
36
Transaction Identifier 1
Transaction State 1
Message Identifier 1
Time stamp
Transaction Identifier 2
Transaction State 2
Message Identifier 2
Time stamp
Transaction Identifier n
Transaction State n
Message Identifier n
Time stamp

FIG. 8/1

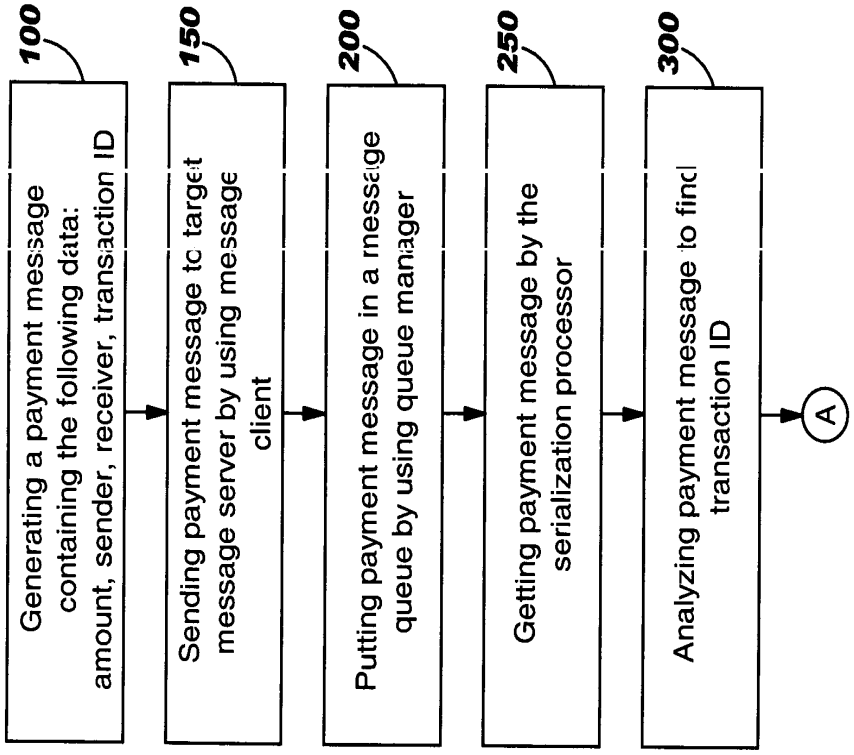
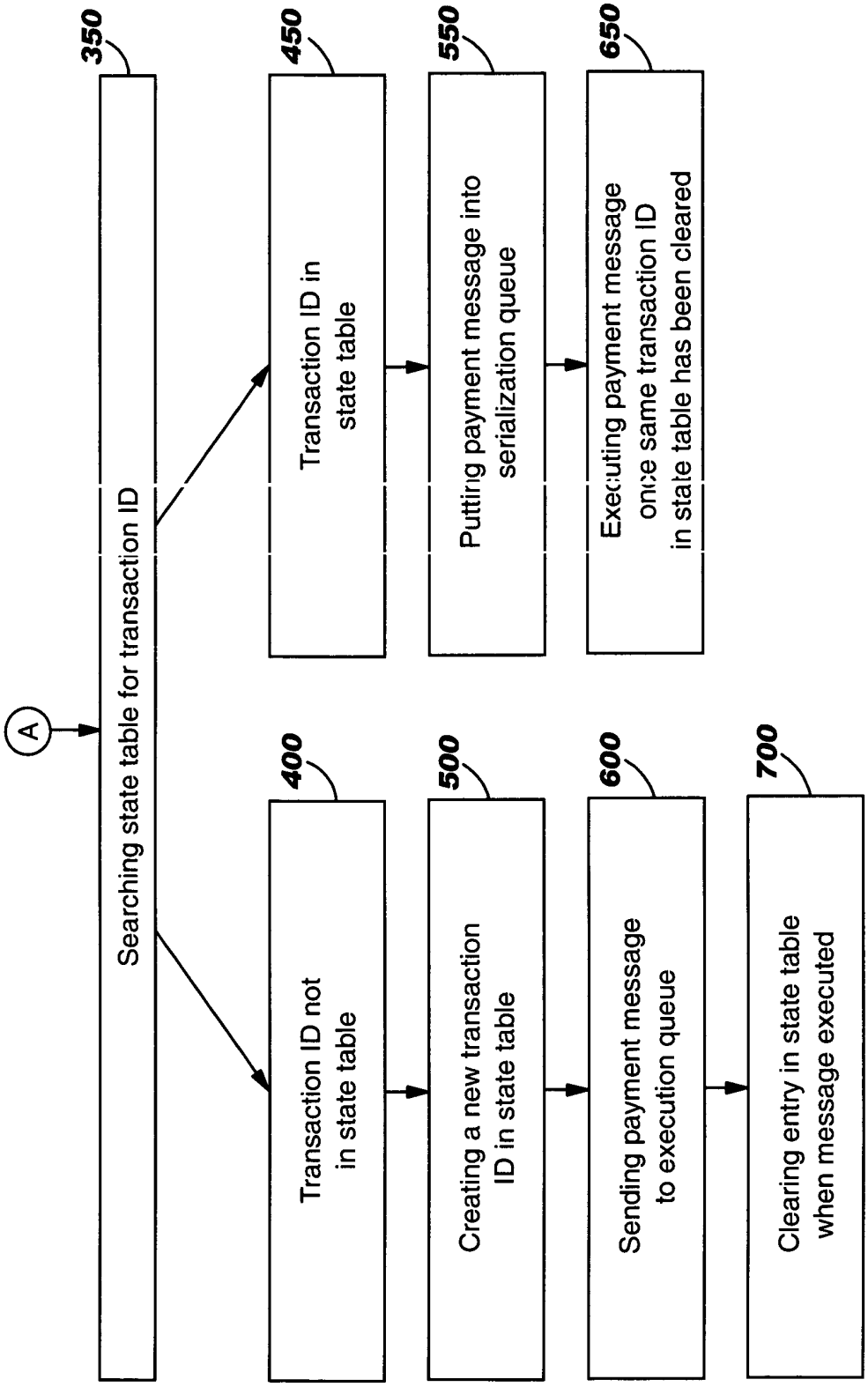

100
Generating a payment message containing the following data: amount, sender, receiver, transaction ID
150
Sending payment message to target message server by using message client
200
Putting payment message in a message queue by using queue manager
250
Getting payment message by the serialization processor
300
Analyzing payment message to find transaction ID
A

FIG. 8/2


A
Searching state table for transaction ID
350
Transaction ID not in state table
400
Creating a new transaction ID in state table
500
Sending payment message to execution queue
600
Clearing entry in state table when message executed
700
Transaction ID in state table
450
Putting payment message into serialization queue
550
Executing payment message once same transaction ID in state table has been cleared
650